Appendix of Straighten Viscous Rectified Flow via Noise Optimization

Jimin Dai[1], Jiexi Yan[2], Jian Yang[1], Lei Luo[1]
[1]PCA Lab, Nanjing University of Science and Technology, [2]Xidian University
{jimindai, csjyang}@njust.edu.cn, {jxyan1995, luoleipitt}@gmail.com

A. Proof

In this section, we give some definitions and proofs of theorems mentioned in the main paper.

Definition 2. *(X_0, X_1) is called an arbitrary coupling when the noise X_0 and the image X_1 are randomly sampled and randomly matched.*

Definition 3. *(X_0, X_1) is called a deterministic coupling if the noise X_0 is randomly sampled, and the image X_1 is generated by a pre-trained model that uses the noise X_0 as the input.*

Theorem 1. *In $(n \times n)$-dimensional space, for straight-line interpolation trajectories $X^{(\cdot)} = \left\{X_t^{(\cdot)} : t \in [0,1]\right\}$, the probability of $X^{(i)}$ and $X^{(j)}$ crossing at point X_t at time step t is $P \sim O(e^{-c(n\times n)}), c > 0$.*

Proof. In an $(n \times n)$-dimensional space, two distinct trajectories $X^{(\cdot)} = \left\{X_t^{(\cdot)} : t \in [0,1]\right\}$ crossing at the same point X_t at time t can be represented as:

$$tX_1^{(i)} + (1-t)X_0^{(i)} = tX_1^{(j)} + (1-t)X_0^{(j)},$$

i.e.

$$t = \frac{X_0^{(i)} - X_0^{(j)}}{X_0^{(i)} - X_0^{(j)} - (X_1^{(i)} - X_1^{(j)})}.$$

As t is a scalar and X_t is an $(n \times n)$-dimensional matrix, for the above equation to hold, the following consistency condition must be satisfied:

$$t = \frac{X_{0,(k,l)}^{(i)} - X_{0,(k,l)}^{(j)}}{X_{0,(k,l)}^{(i)} - X_{0,(k,l)}^{(j)} - (X_{1,(k,l)}^{(i)} - X_{1,(k,l)}^{(j)})}, \quad k,l = 0,1,\ldots,n-1,$$

where (k,l) denotes the coordinate of the $(n \times n)$-dimensional matrix, the consistency condition requires that the value of each component in the matrix is equal to t. Assume that the probability of the component at coordinate (k,l) being equal to t is $P \in [0,1]$, the probability of satisfying the consistency condition would be $P_{constant} = p^{(n\times n)} = e^{(n\times n)\ln(p)}$.

Given that $P \in [0,1]$, and $\ln(p) < 0$, let $P_{constant} = e^{-c(n\times n)}, c > 0$. Due to the independence assumption (which is commonly made in generative models, where the modeling of each dimension is assumed to be independent), $P_{constant}$ decreases exponentially with $(n \times n)$, so $P_{constant} \sim O(e^{-c(n\times n)})$.

Theorem 2. *For each intermediate state X_t along a PF trajectory, the velocity differences between X_t are greater than their state differences:*

$$\Delta({v_{ref}}^{(i)}, {v_{ref}}^{(j)}) \geq \Delta({X_t}^{(i)}, {X_t}^{(j)}),$$

where $\Delta(\cdot,\cdot) = \mathbb{E}[\|\cdot - \cdot\|_F^2]$ and $\mathbb{E}[\cdot]$ is the expectation.

Proof. In the context of flow matching models, we introduce the Frobenius norm to measure the difference between data points. Since the training involves randomness, we introduce the expectation to quantify the "average difference" between data points. To ensure that the metric result is non-negative, we define $\Delta(\cdot,\cdot)$ as the expectation of the squared Frobenius norm, i.e. $\Delta(\cdot,\cdot) = \mathbb{E}[\|\cdot - \cdot\|_F^2]$.

Given two noise-image data pairs $({X_0}^{(i)}, {X_1}^{(i)})$ and $({X_0}^{(j)}, {X_1}^{(j)})$, where ${X_0}^{(i)}, {X_0}^{(j)} \sim N(0, I)$ and ${X_1}^{(i)}; {X_1}^{(j)}$ are samples from the image distribution. The intermediate states obtained by straight-line interpolating them are ${X_t}^{(i)} = t{X_1}^{(i)} + (1-t){X_0}^{(i)}$ and ${X_t}^{(j)} = t{X_1}^{(j)} + (1-t){X_0}^{(j)}$, and the difference between the two intermediate states is represented as:

$$\begin{aligned}
\Delta({X_t}^{(i)}, {X_t}^{(j)}) &= \mathbb{E}\left[\left\|{X_t}^{(i)} - {X_t}^{(j)}\right\|_F^2\right] \\
&= \mathbb{E}\left[\left\|t({X_1}^{(i)} - {X_1}^{(j)}) + (1-t)({X_0}^{(i)} - {X_0}^{(j)})\right\|_F^2\right] \\
&= t^2\mathbb{E}\left[\left\|{X_1}^{(i)} - {X_1}^{(j)}\right\|_F^2\right] + (1-t)^2\mathbb{E}\left[\left\|{X_0}^{(i)} - {X_0}^{(j)}\right\|_F^2\right] + 2t(1-t)\mathbb{E}\left[Tr(({X_1}^{(i)} - {X_1}^{(j)})^T({X_0}^{(i)} - {X_0}^{(j)}))\right].
\end{aligned}$$

Since ${X_0}^{(i)}$ and ${X_0}^{(j)}$ are independent, the cross term is 0. Therefore, we simplify to:

$$\Delta({X_t}^{(i)}, {X_t}^{(j)}) = t^2\mathbb{E}\left[\left\|{X_1}^{(i)} - {X_1}^{(j)}\right\|_F^2\right] + (1-t)^2\mathbb{E}\left[\left\|{X_0}^{(i)} - {X_0}^{(j)}\right\|_F^2\right].$$

Given that ${X_0}^{(i)}, {X_0}^{(j)} \sim N(0, I)$, we have

$$\begin{aligned}
\mathbb{E}[\|X_0^{(i)} - X_0^{(j)}\|_F^2] &= \mathbb{E}\left[\sum_{k=1}^{n}\sum_{l=1}^{n}(X_{0,(k,l)}^{(i)} - X_{0,(k,l)}^{(j)})^2\right] \\
&= \sum_{k=1}^{n}\sum_{l=1}^{n}\mathbb{E}\left[(X_{0,(k,l)}^{(i)} - X_{0,(k,l)}^{(j)})^2\right] \\
&= \sum_{k=1}^{n}\sum_{l=1}^{n}\left\{\text{Var}(X_{0,(k,l)}^{(i)} - X_{0,(k,l)}^{(j)}) + \left(\mathbb{E}\left[X_{0,(k,l)}^{(i)} - X_{0,(k,l)}^{(j)}\right]\right)^2\right\} \\
&= \sum_{k=1}^{n}\sum_{l=1}^{n}\left[\text{Var}(X_{0,(k,l)}^{(i)}) + \text{Var}(X_{0,(k,l)}^{(j)})\right] \\
&= 2n^2,
\end{aligned}$$

where $n \times n$ is the dimension of X_0, $X_{0,(k,l)}$ represents the element value of X_0 at the coordinate (k, l) . so we get

$$\Delta({X_t}^{(i)}, {X_t}^{(j)}) = t^2\Delta({X_1}^{(i)}, {X_1}^{(j)}) + 2n^2(1-t)^2.$$

The constant velocity fields for these two sample pairs are ${v_{ref}}^{(i)} = {X_1}^{(i)} - {X_0}^{(i)}$ and ${v_{ref}}^{(j)} = {X_1}^{(j)} - {X_0}^{(j)}$, and the difference between the two velocity fields is represented as:

$$\Delta({v_{ref}}^{(i)}, {v_{ref}}^{(j)}) = \mathbb{E}\left[\left\|{X_1}^{(i)} - {X_0}^{(i)} - ({X_1}^{(j)} - {X_0}^{(j)})\right\|_F^2\right],$$

Similarly, since ${X_0}^{(i)}, {X_0}^{(j)} \sim N(0, I)$, we have

$$\begin{aligned}
\Delta({v_{ref}}^{(i)}, {v_{ref}}^{(j)}) &= \mathbb{E}\left[\left\|{X_1}^{(i)} - {X_1}^{(j)}\right\|_F^2\right] + \mathbb{E}\left[\left\|{X_0}^{(i)} - {X_0}^{(j)}\right\|_F^2\right] \\
&= \Delta({X_1}^{(i)}, {X_1}^{(j)}) + 2n^2
\end{aligned}$$

As $t \in [0, 1]$, we have:

$$\begin{aligned}
&\Delta({v_{ref}}^{(i)}, {v_{ref}}^{(j)}) - \Delta({X_t}^{(i)}, {X_t}^{(j)}) \\
=&(1-t^2)\Delta({X_1}^{(i)}, {X_1}^{(j)}) + 2n^2(2t - t^2) \\
\geq&0,
\end{aligned}$$

so we can get $\Delta({v_{ref}}^{(i)}, {v_{ref}}^{(j)}) \geq \Delta({X_t}^{(i)}, {X_t}^{(j)})$.

Definition 4. *We denote that X is rectifiable if v^X (denote neural velocity field train by X) is locally bounded and the solution to the integral equation of the form*

$$Z_t = Z_0 + \int_0^t v^X(Z_t, t, v^X_{t-\Delta t})dt, \ \ \forall t \in [\Delta t, 1], \ \ v^X_0 = 0, \ \ Z_0 = X_0$$

exists and is unique. In this case, $Z = \{Z_t : t \in [0, 1]\}$ is called the viscous rectified flow induced by X.

Theorem 3. *Assume X is rectifiable and Z is its viscous rectified flow. The marginal law of Z_t equals that of X_t at every time t, i.e., Law(Z_t) = Law(X_t), $\forall t \in [0, 1]$.*

Proof. The velocity field of the viscous rectified flow, after introducing the historical velocity term, is defined as:

$$v_\theta(X_t, t, h_t) = \mathbb{E}[\dot{X}_t | X_t = x_t],$$

where x_t represents the specific sample value of X_t, $h_t = v_{t-\Delta t}$, $\dot{X}_t = \partial_t X_t$ represents the time derivative. After the PF begins to evolve, the historical velocity term $v_{t-\Delta t}$ transitions from 0 to the conditional distribution:

$$P(h_t | v_{\theta,t-\Delta t}) = \delta(h_t - v_{\theta,t-\Delta t}),$$

indicating that at time step t, the historical velocity is deterministic and corresponds to the velocity predicted by the velocity field from the previous time step. Therefore, the joint distribution of X_t can be represented as:

$$\bar{\pi}_t(x_t, h_t) = \pi_t(x_t)P(h_t | v_{\theta,t-\Delta t}).$$

Assume $\varphi(x_t, h_t)$ is a smooth test function that satisfies appropriate boundary conditions. We integrate it as follows:

$$I(t) = \int \varphi(x_t, h_t)\bar{\pi}_t(x_t, h_t)dx_t dh_t.$$

From $\bar{\pi}_t(x_t, h_t) = \pi_t(x_t)P(h_t | v_{\theta,t-\Delta t})$, we obtain

$$I(t) = \int \varphi(x_t, h_t)\pi_t(x_t)\delta(h_t - v_{\theta,t-\Delta t})dx_t dh_t.$$

Using the properties of the Dirac delta function, the integral over the h_t part can be simplified to:

$$I(t) = \int \varphi(x_t, v_{\theta,t-\Delta t})\pi_t(x_t)dx_t.$$

Then, differentiate $I(t)$ with respect to time t, and we get:

$$\frac{d}{dt}I(t) = \frac{d}{dt}\int \varphi(x_t, v_{\theta,t-\Delta t})\pi_t(x_t)dx_t.$$

Since $\varphi(x_t, v_{\theta,t-\Delta t})$ only depends on x_t and $v_{\theta,t-\Delta t}$, where $v_{\theta,t-\Delta t}$ s the velocity at the previous time step and does not directly depend on time t, we can apply the chain rule to compute the derivative of $\pi_t(x_t)$ with respect to time:

$$\frac{d}{dt}I(t) = \int \varphi(x_t, v_{\theta,t-\Delta t})\partial_t\pi_t(x_t)dx_t.$$

Next, consider the advection term $\nabla_{x_t} \cdot (v_\theta(x_t, t, h_t)\bar{\pi}_t(x_t, h_t))$. Similarly, substitute $\bar{\pi}_t(x_t, h_t) = \pi_t(x_t)\delta(h_t - v_{\theta,t-\Delta t})$ and integrate with the test function:

$$J(t) = \int \nabla_{x_t} \cdot (v_\theta(x_t, t, h_t)\pi_t(x_t)\delta(h_t - v_{\theta,t-\Delta t}))\, \varphi(x_t, h_t)dx_t dh_t.$$

First, integrate h_t:

$$J(t) = \int \nabla_{x_t} \cdot (v_\theta(x_t, t, v_{\theta,t-\Delta t})\pi_t(x_t))\, \varphi(x_t, v_{\theta,t-\Delta t})dx_t.$$

This expression indicates that the convection term depends only on $v_\theta(x_t, t, v_{\theta,t-\Delta t})$ and $\pi_t(x_t)$, while the historical velocity h_t is fixed as $v_{\theta,t-\Delta t}$ by the Dirac delta function.

By adding the time derivative and the convection term, we obtain the continuity equation:

$$\frac{d}{dt}I(t) + J(t) = \int \varphi(x_t, v_{\theta,t-\Delta t})\partial_t \pi_t(x_t) dx_t + \int \nabla_{x_t} \cdot \left(v_\theta(x_t, t, v_{\theta,t-\Delta t})\pi_t(x_t)\right) \varphi(x_t, v_{\theta,t-\Delta t}) dx_t.$$

Based on the properties of the continuity equation $\frac{d}{dt}I(t) + J(t) = 0$, we have:

$$\int \varphi(x_t, v_{\theta,t-\Delta t}) \left(\partial_t \pi_t(x_t) + \nabla_{x_t} \cdot \left(v_\theta(x_t, t, v_{\theta,t-\Delta t})\pi_t(x_t)\right)\right) dx_t = 0.$$

Since $\varphi(x_t, h_t)$ is an arbitrary smooth test function and the integral over x_t and h_t spans the entire space, the expression inside the integral must be zero, i.e.,

$$\partial_t \pi_t(x_t) + \nabla_{x_t} \cdot \left(v_\theta(x_t, t, v_{\theta,t-\Delta t})\pi_t(x_t)\right) = 0.$$

Therefore, by introducing the test function $\varphi(x_t, h_t)$, we have proven that under the condition $\bar{\pi}_t(x_t, h_t) = \pi_t(x_t)\delta(h_t - v_{\theta,t-\Delta t})$, X_t still satisfies the continuity equation:

$$\partial_t \bar{\pi}_t(x_t, h_t) + \nabla_{x_t} \cdot \left(v^X(x_t, t, h_t)\bar{\pi}_t(x_t, h_t)\right) = 0.$$

During inference, because Z_t is driven by the same velocity field v^X, its marginal distribution Law(Z_t) will solve exactly the same equation under the same initial conditions $Z_0 = X_0$. Since, under our setup, both Z_0 and X_0 follow the same distribution (whether sampled directly from the standard Gaussian distribution or from the noise distribution optimized via reparameterization), and based on the uniqueness theory [3] of the solution to the continuity equation, we conclude that their marginal distributions are identical:

$$\text{Law}(Z_t) = \text{Law}(X_t), \quad \forall t \in [0, 1].$$

B. Additional Results

In this section, we present additional experimental results. Fig.I provides a visualization of the inference trajectories for both RF and VRFNO on synthetic data, further validating the effectiveness of VRFNO in straightening the trajectories. Fig.II to Fig.IV show the qualitative results generated by VRFNO on the CIFAR-10 dataset at different inference steps, compared with the RF method. As VRFNO's generated images are influenced by the encoder-extracted prior information from real images, and different models generate slightly different images for the same noise, we cannot use identical noise in VRFNO and other models (Other models do not involve the use of real image prior information) to produce comparable images for a direct comparison. Nevertheless, visual inspection clearly demonstrates that VRFNO, as an alternative to Reflow, achieves performance comparable to or even superior to that of RF.

Tab.X1 compares the model parameter counts required by VRFNO and other methods. VRFNO introduces a new joint framework of an encoder and neural velocity field on top of RF. Although it requires training an additional encoder compared to RF, the parameter count required by the encoder is much smaller than that of the neural velocity field, so it does not impose a significant resource burden. CAF improves RF by proposing to use the neural initial velocity field and the neural acceleration field for joint inference, where the neural initial velocity field provides the estimated initial velocity to the neural acceleration field, offering auxiliary information about the direction of the PF. This means that it requires training two neural velocity fields. These two fields have the same architecture and are used as the neural initial velocity field and the neural acceleration field, respectively. Compared to VRFNO, CAF incurs a higher resource cost. Notably, the model architecture of CAF is different from that of RF, while our method adopts the model architecture of RF. Therefore, in experiments with different resolutions, VRFNO uses the same number of parameters for the neural velocity field as RF. However, in experiments with resolutions of 64×64 and above, we follow RF and change the model architecture, resulting in a decrease in the number of parameters.

Tab.X2 provides experimental evidence supporting the analysis of Reflow in Section 2.2. Diffusion models typically exhibit rapid convergence in the early stages of training, followed by an extended plateau period during which their generative capabilities slowly improve. Therefore, the initial phase of training provides an intuitive window for observing the learning behavior of models under different data and training strategies. Based on this, we select the first 5,000 training iterations as the analysis interval to investigate the impact of different training data and strategies on model performance. Specifically,

Table X1. Comparison of parameters of each method

Method	Resolution	Parameters of model-1(M)	Parameters of model-2(M)
RF	32×32	Neural velocity field(64.4)	-
RF	64×64	Neural velocity field(28.6)	-
CAF	32×32	Neural initial velocity field(55.7)	Neural acceleration field(55.7)
CAF	64×64	Neural initial velocity field(295.9)	Neural acceleration field(295.9)
VRFNO	32×32	Encoder(2.4)	Neural velocity field(64.4)
VRFNO	64×64	Encoder(5.0)	Neural velocity field(28.6)
VRFNO	128×128	Encoder(15.3)	Neural velocity field(80.2)
VRFNO	256×256	Encoder(15.5)	Neural velocity field(90.9)

Table X2. Comparison of data volume and number of data reuse

Coupling	Data volume	Number of data usage	Training iterations	FID(↓)
Deterministic coupling	100000	1	200	425.90
		5	1000	207.05
		25	5000	45.16
	500000	1	1000	218.29
		2	2000	110.45
		5	5000	48.43
Arbitrary coupling	500000	1	1000	355.27
		5	5000	347.61
	-	1	5000	347.69

we employ the following four configurations to train the Neural Velocity Field from scratch: (1) 100,000 deterministic couplings; (2) 500,000 deterministic couplings; (3) 500,000 arbitrary couplings; (4) Unlimited arbitrary couplings (where noise is dynamically sampled and randomly paired with real images during training, rather than using a pre-stored finite set of coupled data). In the experimental setup, the batch size is fixed at 500, and the NFE is set to 1. We ensure that the total amount of data encountered by all models from the start of training up to the current training iteration is identical. This eliminates the influence of data volume on the results, allowing us to focus solely on the differences in data and strategies.

Through experiments and analysis, we find that the success of Reflow can be attributed, but is not limited, to the reuse of deterministic couplings. As shown in Tab.X2, when the number of training iteration is 1,000, the training with 100,000 data points reuses the data 5 times, while the training with 500,000 data points has only traversed the training data once. Based on the FID score, the training with 100,000 data points yields better results. Similarly, when the number of training iteration is 5,000, the training with 100,000 data points reuses the data 25 times, while the training with 500,000 data points has only traversed the training data 5 times. Again, the training with 100,000 data points performs better according to the FID score. We also compared the difference between training with and without data reuse when using arbitrary couplings. As shown in the tab.X2, there is almost no difference between the two. Additionally, we compared the results of training with deterministic couplings and arbitrary couplings when data is not reused, as seen in the 4th and 7th rows of the tab.X2. The results indicate that using deterministic couplings improves the image quality during fast sampling. Therefore, we conclude that the effectiveness of Reflow relies on two key factors: (1) the use of deterministic couplings as training data and (2) data reuse. However, data reuse only proves effective when training with deterministic couplings, which aligns with the distillation strategy's mechanism mentioned in the main paper.

The main paper mentions that there is a distribution gap between the images in the deterministic couplings and the real images in the dataset. This distribution gap causes error accumulation during Reflow training, leading to a gradual decline in the image quality generated by each generation of the model, while the inference trajectory becomes progressively straighter. This phenomenon can be clearly observed in Tab.X3, for the first to third generations of the RF, when using the RK45 adaptive sampler, the sampling steps decrease progressively, and the quality of the generated images also correspondingly declines.

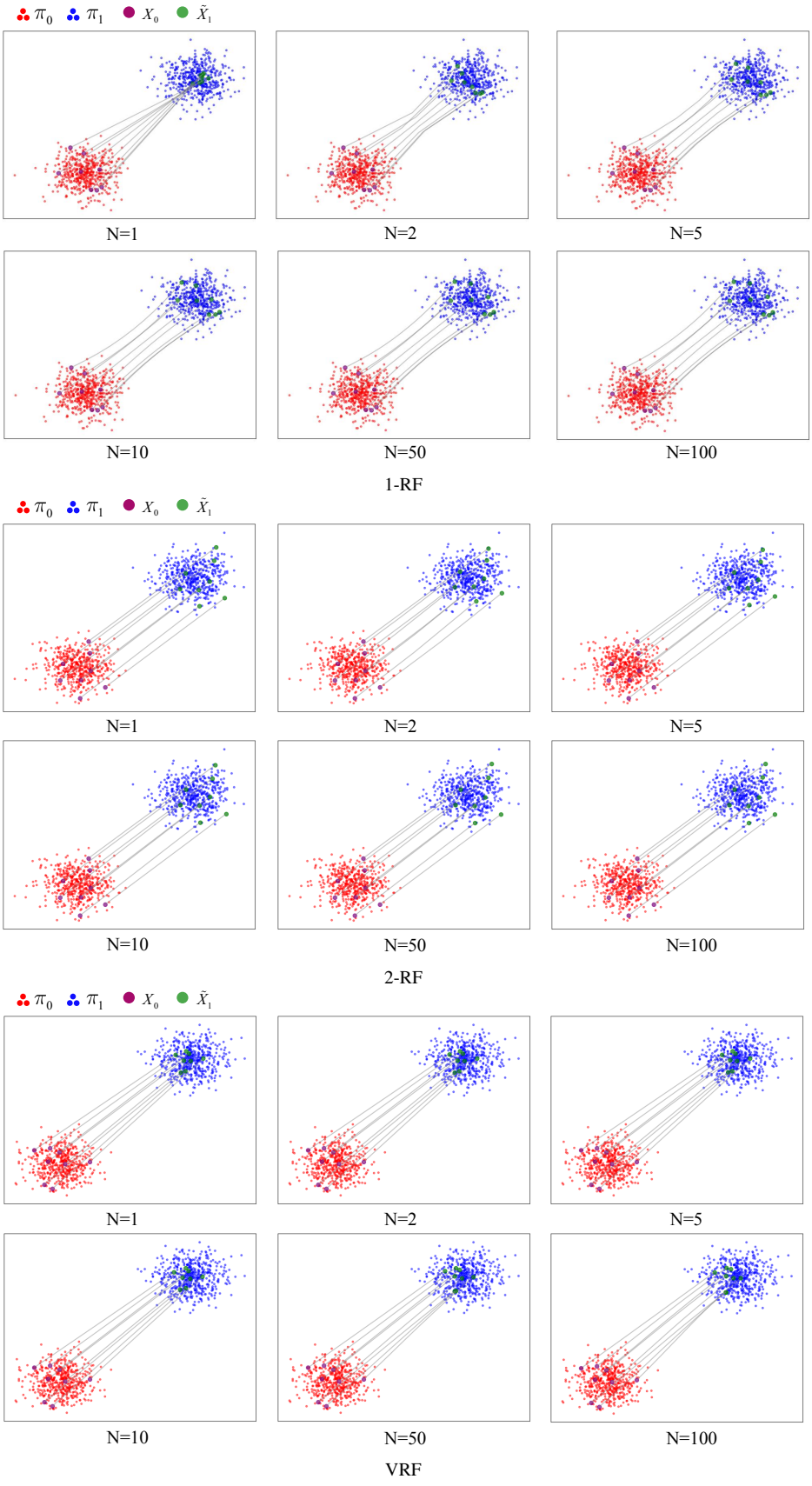

Figure I. **Visualization of inference trajectories of different models on synthetic data.**

(a) 1-RF (b) 2-RF (c) CAF (c) VRFNO

Figure II. **Qualitative results of different models on CIFAR-10 (I).** Image generation with 1 step

(a) 1-RF (b) 2-RF (c) CAF (c) VRFNO

Figure III. **Qualitative results of different models on CIFAR-10 (II).** Image generation with 5 steps

(a) 1-RF (b) 2-RF (c) CAF (c) VRFNO

Figure IV. **Qualitative results of different models on CIFAR-10 (III).** Image generation with 10 steps

Table X3. Performance comparison among N-RFs

N-RF	NFE	IS(↑)	FID(↓)
1-RF	127	9.60	2.58
2-RF	110	9.24	3.36
3-RF	104	9.01	3.96

C. Implementation of Reflow

RF attributes the curvature of inference trajectories to the model's loss function being defined to minimize the expected difference between the predicted and the ground truth velocity: at the crossing point of trajectories, the model's prediction becomes the average velocity of all intermediate states that pass through the crossing point. This averaging effect results in a curved trajectory, causing the model to lose the desirable properties of a straight flow. To straighten the PF trajectory and enable few-step or single-step generation, RF introduces Reflow.

Reflow samples Z_0 from the initial distribution π_0 and inputs it into a pre-trained model (referred to as the "1st generation model") to generate a corresponding sample Z_1 that approximates the target distribution. These pairs (Z_0, Z_1) are called *deterministic coupling*. In a deterministic PF, the sampling process can be formalized as an initial value problem for the ODE. Due to the uniqueness of the ODE solution, the trajectories between these *deterministic couplings* are not crossing. However, these trajectories are usually not straight lines. Therefore, Reflow performs straight-line interpolation on these deterministic couplings to obtain new straight-line trajectories, which are then used to fine-tune the model. (Note that, although the inferred curved paths between *deterministic couplings* are not crossing, the straight-line trajectories obtained by performing straight-line interpolation on these *deterministic couplings* may still have a few crossing points.)

The newly fine-tuned model (referred to as the "2nd generation model") can be viewed as a neural velocity field trained on straight-line trajectories with a small number of crossing points (fewer than the crossing points between straight-line trajectories from *arbitrary couplings*). The trajectories between the *deterministic couplings* generated by the 2nd generation model will be straighter compared to those from the 1st generation model, and similarly, no crossing points will occur. To make the generated trajectories even closer to straight lines, this process can be repeated. *Deterministic couplings* generated by 2nd generation model can be used to fine-tune and obtain the 3rd generation model, and so on.

D. Experiment Configuration

Our experiments follow the experimental setup and model framework of RF. Additionally, the encoder training does not use exponential moving average for smoothing. For the newly added hyperparameters, α is set to 0.0000001, Δt is set to 0.01, and t is randomly sampled from $[\Delta t, 1]$ with intervals of Δt.

E. Encoder and Noise Optimization

E.1. Architecture of Encoder

In the main paper, we mentioned that our encoder is refer to the encoder architecture in VAE [2]. Fig.V shows our encoder architecture, which is an improvement upon the original VAE's encoder to introduce randomness. Specifically, after the encoder extracts features and before these features are passed through the fully connected layer to obtain the mean and variance, we introduce Gaussian noise and fuse it with the features using a random mask. This approach ensures that when different noises match the same image, the means and variances used for optimization differ. This difference not only makes the model more robust but also ensures the randomness of subsequent image generation.

E.2. Noise Optimization

In Section 3.2, we discussed the differences between our noise optimization method and the traditional approach, with its schematic diagram shown in Fig.VI. Specifically, the traditional noise optimization method typically iterates and optimizes based on the evaluation score from the reward model, using gradient updates, and usually requires multiple iterations of updates. For instance, [1] requires up to 50 iterations of updates to obtain the final optimized noise. In contrast, our noise optimization method leverages prior information from the existing dataset and obtains the optimized noise through 2 matrix operations. The specific implementation involves the encoder outputting two matrices, which are then multiplied and added with the original noise, respectively. This optimization method is formally equivalent to the reparameterization technique, which is why we refer to it as using reparameterization technology for noise optimization.

Our method leverages the prior information provided by the dataset (i.e., the mean matrix and variance matrix), which eliminates the need for iterative optimization starting from the initial random noise.Typically, prior information represents the statistical properties of the data, offering an effective direction for noise optimization, allowing the noise to quickly converge to an approximate optimal solution. Additionally, by employing reparameterization techniques, we transform the noise optimization problem into multiplication and addition operations involving the noise matrix, variance matrix, and mean matrix.This approach is, in form, equivalent to directly reconstructing the statistical properties of the noise in certain cases,

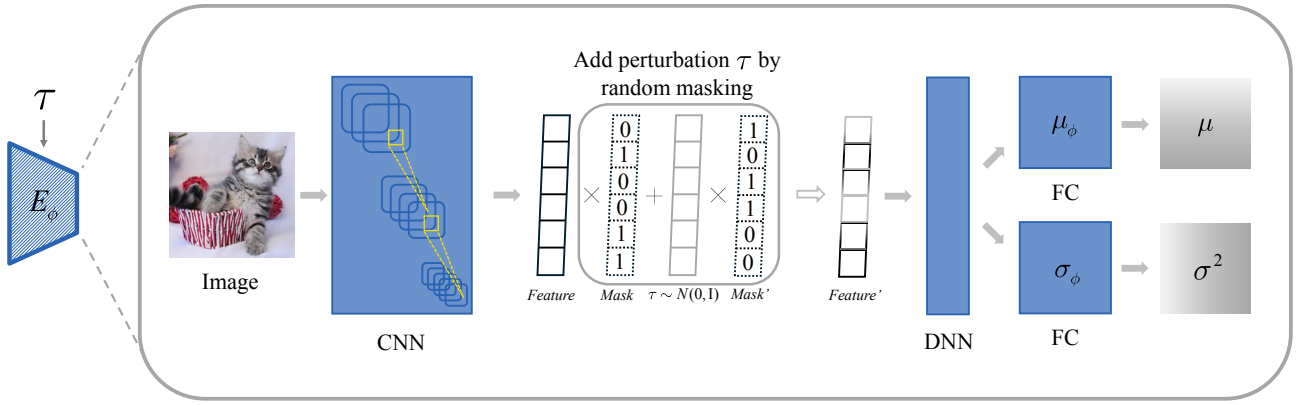

Figure V. **Architecture of encoder.**

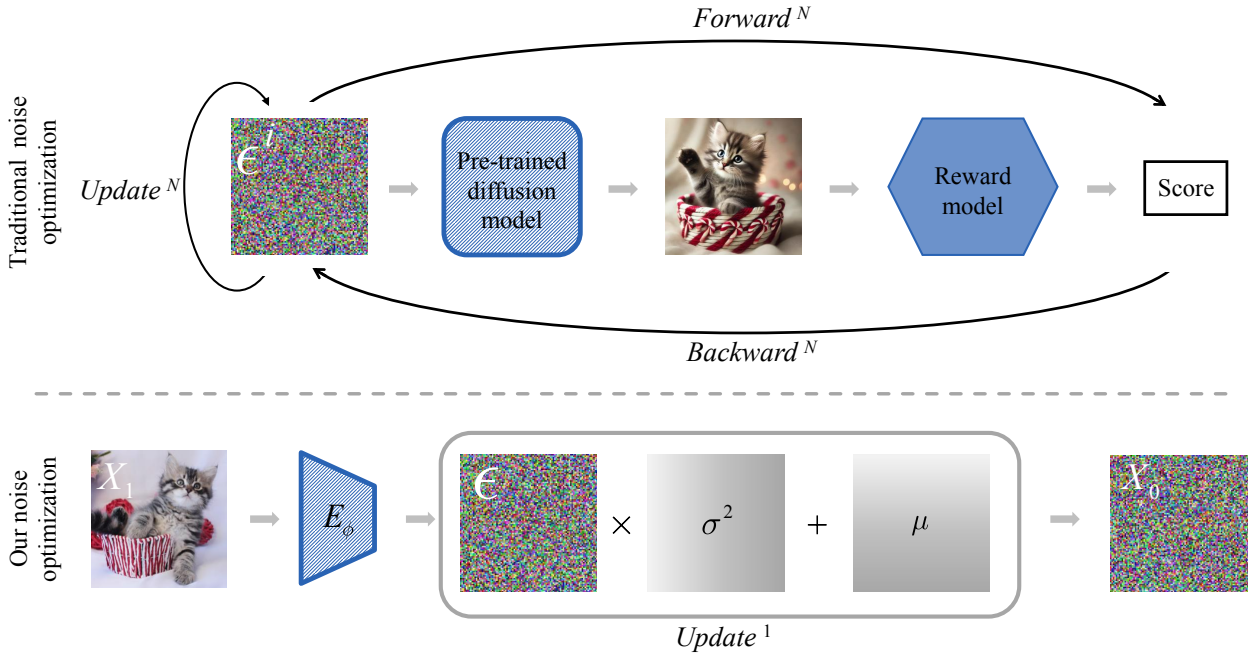

Figure VI. **Schematic diagram comparing the traditional noise optimization method with ours.**

without the need to iteratively update each noise component. Therefore, in practical applications, only one iteration is needed to achieve the optimization goal.

During the sampling process, although prior information from the images in the dataset is used to optimize the noise, it does not affect the diversity of the generated images. As shown in Fig.VII, the left side shows images randomly sampled from the dataset, which are used to input the encoder and generate the mean and variance matrices for noise optimization. The right side shows the images generated by the noise optimized using the information from the left-side images. The similarity between these images varies, maintaining the diversity of image generation.

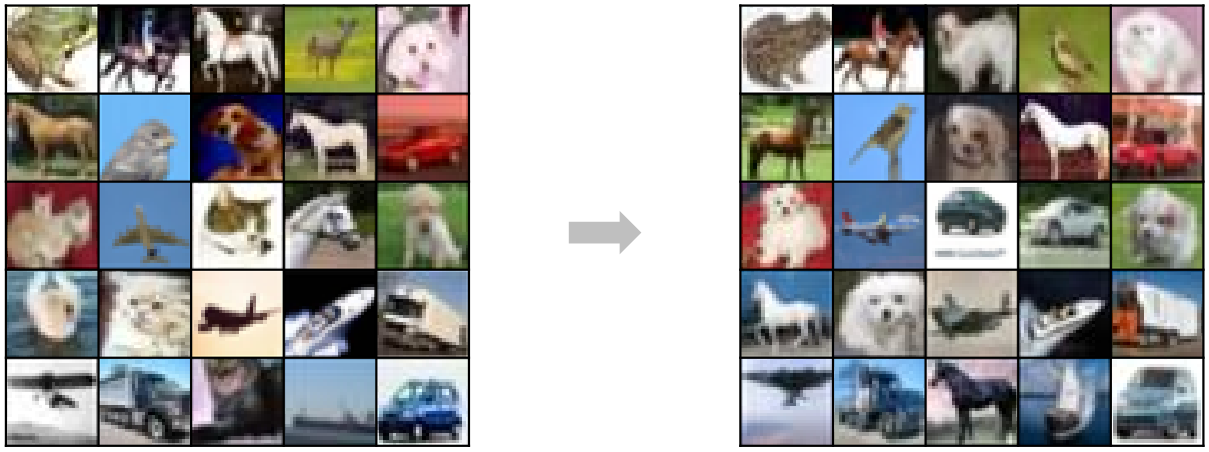

Figure VII. **Images taken from the dataset for optimizing noise and the corresponding generated images.**

F. Application in Downstream

The mechanism and concept of VRFNO can also be applied to downstream applications to further enhance the performance of related models on specific tasks. For example, [5] uses a conditional adapter to fuse the watermark with the image. Training the adapter with feedback from the main model—similar to VRFNO—may improve the generation quality. MACS in [7] requires multi-step generation. Incorporating the VRFNO mechanism—using audio embeddings to optimize the noise—may accelerate generation. [4] can use the VRFNO training framework by introducing a noise modification module jointly fine-tuned with the main model. This module filters out phrases to be erasured and embeds the remaining phrases into the noise for subsequent generation. [6] can learn a noise subspace for each image as discussed in VRFNO, enabling noise sampling from this subspace during training and avoiding the computational overhead of sorting and regrouping samples and noise within a batch.

References

[1] Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. Reno: Enhancing one-step text-to-image models through reward-based noise optimization. *Advances in Neural Information Processing Systems*, 37:125487–125519, 2025. 8

[2] Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017. 8

[3] Thomas G Kurtz. Equivalence of stochastic equations and martingale problems. *Stochastic analysis 2010*, pages 113–130, 2011. 4

[4] Shilin Lu, Zilan Wang, Leyang Li, Yanzhu Liu, and Adams Wai-Kin Kong. Mace: Mass concept erasure in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6430–6440, 2024. 10

[5] Shilin Lu, Zihan Zhou, Jiayou Lu, Yuanzhi Zhu, and Adams Wai-Kin Kong. Robust watermarking using generative priors against image editing: From benchmarking to advances. *arXiv preprint arXiv:2410.18775*, 2024. 10

[6] Xu Shifeng, Yanzhu Liu, and Adams Wai-Kin Kong. Easing training process of rectified flow models via lengthening inter-path distance. In *The Thirteenth International Conference on Learning Representations*. 10

[7] Hao Zhou, Xiaobao Guo, Yuzhe Zhu, and Adams Wai-Kin Kong. Macs: Multi-source audio-to-image generation with contextual significance and semantic alignment. *arXiv preprint arXiv:2503.10287*, 2025. 10